Information Processing Capability and Inter-Organizational Collaboration in Software development

Mojgan Mutashami and Cheickna Sylla*

Abstract

This paper reports an empirical investigation focusing on the effects of the information processing on the performance of inter-organizational collaboration. Inter-organizational information processing addresses the information exchange, as well as information utilization across organizational boundaries. Using these, and other factors associated with information processing capability (IPC) and good management practices, the paper proposes and tests several important hypotheses confirming how a higher level of inter-organizational IPC relates to a higher level of software development performance. The results also depict the significant contributors to the information processing capacity in a collaborative software development.

Keyword: Software development, Inter-organizational collaboration

1. Introduction

Advances in technology, along with globalizations, have made inter-organizational alliances a norm in modern day business. There is a considerable lack of knowledge, however, with regards to the characteristics, business requirements and proper conduct of business when it comes to inter-organizational collaboration [1]. Inter-organizational collaborative software development (ICSD) as one specific form of group collaboration entails inter-organizational alliances, multi-team collaboration working for various functional organizations, and often no clear central authority. Such software development practices cross national, linguistic,organizational, and cultural boundaries. ICSD projects are often large system development undertakings with a very high price tag.

Past studies indicate that effective information exchange between collaborating teams is critical for the success of collaborative software development [2]. One of the most important determinants of effectiveness of information exchange is how capable the collaborating teams are in information processing and communications. The study presented in this paper addresses the critical role of information flow and communication in general and software development management in particular when two or more companies work collaboratively in developing software. The concept of information processing capacity (IPC) is developed and the role of IPC is tested in ICSD context. The paper reports a large comprehensive empirical research investigation involving over 50 companies experienced in software development.

*School of Management New Jersey Institute of Technology, USA

2. Inter-organizational Collaborative Software Development (ICSD)

ICSD is characterized by a large scale, complex, dynamic and large pool of diverse users, multi-resources, and highly interdependent components [3]. All these make ICSD an uncertain and high-risk endeavor [4][5]. Kraut and Streeter [3] identify characteristics of ICSD and large-scale software development projects as followings:

- Each software development project is unique, and there is no past history of success or failure to learn from.
- Software development specifications are invariably incomplete, in consequence of limited domain knowledge and division of labor.
- Software development involves non-routine activities with no existing prototypes, application or system to modify or change.
- There is a large set of stakeholders with various, and at times contradictory, points of view.
- As users' and/or markets' requirements change over time, managers and developers run into obstacles, and new criteria surface, and the governing constraints and requirements of the software change.

Such attributes change and extend the domain and nature of software development management. Inter-organizational collaboration has introduced further process complexity and uncertainty, to the already troubled, complex, and inter-related activities of large scale software development projects. Management of such uncertain and complex projects demands close collaboration between collaborating teams. Communication, coordination and collaboration are collectively the key success factors of joint alliances [6][7][8]. Even though various software engineering works have addressed internal team development management schemes, there is little work on how to coordinate and manage across the activities of dispersed teams and alliances [9].

2.1 Factors influencing the success of ICSD

ICSD that requires collaboration between geographically dispersed organizations, often across multiple countries, is rapidly becoming the norm for technology companies [10]. ICSD projects are highly complex and interdependent [11], which requires group collaboration, precise cross-functional information exchange and cooperation [8][12].

According to [8][13], cross-organizational cooperation and communication thrives on two main objectives: trust and operational alignment. Trust in the context of software development has some unique characteristics. First, trust at the interpersonal (technical) level is somehow more critical. A collaborating environment inherits the problems of traditional software development settings, as discussed in earlier sections. In a collaborative software development endeavor, it is likely that the overall software design and development is distributed among participating agencies. The components developed by various partners are highly interdependent. It is common to blame development partners for dysfunctional components/modules, lack of portability, and lack of compatibility of integrated modules or components. Second, in an ICSD, the type of information exchanged is different. In most collaborative efforts, various types of financial, operational data and information are exchanged, while in an ICSD information pertaining to the design, development, testing, and budgetary data is exchanged. Even though the sensitivity of data is much less critical, the act of communication and coordination is vital to the completion and success of the ICSD activities.

Cultural difference is one of the primary barriers in collaboration efforts. Cultural differences can create cognitive barriers against people's interactions. In ICSD, due to interdependency of the components and highly collaborative nature of the product, such reluctance can be costly, if not fatal. The effect of culture in system development would be even greater in cross-country software development. Developers in different countries may take different approaches to systems development, have different perceptions on end-users, or have different levels of perceived risks [14]. Operation standards may vary by region or by organization. Therefore, a detailed understanding of foreign collaborators' capabilities and practices, as well as local standards and regional infrastructure, is required in collaborative management [15]. Organizations should develop strategies to manage cross-cultural communication, operating guidelines, and language training.

2.2 Information Processing Capacity in ICSD

Effective collaboration, particularly concerning physically dispersed teams such as ICSD, requires accurate and timely information exchange. Kayworth et al. [2] identify effective communication and coordination as a way to control organizational behavior roadblocks. The effectiveness of collaboration, to a considerable degree, is a function of quality, frequency, relevance, integration, and dissemination of information flow or knowledge management [16]. Information flow is determined largely by the quality and adequacy of the communication infrastructure and practices.

ICSD is sensitive to communication and information exchange. It might be beneficial to collaborative design technology that not only focuses on the work tasks but also meets the expectations of the culturally diverse team members [17] and helps with the establishment of trust. Communication technologies offer a set of collaboration functions – meeting same time same place, reviewing and sharing data, application, group memory, scheduling, operation planning over a network, which collectively make up for time and space separation [18].

According to a study on collaborative software development [19], three factors fatal for the success of collaborative information systems are 1) lack of adoptability of the systems so that the coalition cannot respond to changes accordingly, 2) improper design of business partnership, and 3) lack of real time and ad hoc connectivity and sharing opportunity. Other research findings state that many sophisticated collaborative systems are not used extensively because users prefer a less complex, more user-friendly and efficient system [20]. This implies that satisfying user expectation, user-friendliness, and efficiency of the IT supporting communications are important in ICSD.

3. Research Method

In this study uses a two phase model (seen depicted in Figure 1) has been developed to address the effect of influential factors on IPC and subsequently the effect of IPC on a success of ICSD. The first phase is designed to examine the effect of three influential factors on the level of IPC. The first phase identifies the set of factors which affect the nature of communication and information exchange between organizations. These factors are the levels of (1) experience in software development and collaboration, (2) organizational background, (3) management contingency process (known in management literature as "contingency profile"), and (4) information technologies supporting communications. The level of experience can be easily established, but others are not so trivial. Organizational background includes experience in software development and collaboration which refers to the extent to which the ICSD teams have experiences with software development and collaboration. Other items for organizational background can be grouped into four groups (1) Organizational culture, (2) Organization learning capacity, (3) Organization cultural familiarity, and (4) Trust. Sub-constructs, 'Experiences in software development and collaboration' and 'organizational learning capacity' are static or structural variables which are not subject to enhancement or modifications. Level of trust and culture also refer to the current practices in the company which can be improved upon by installment of management contingency approaches.

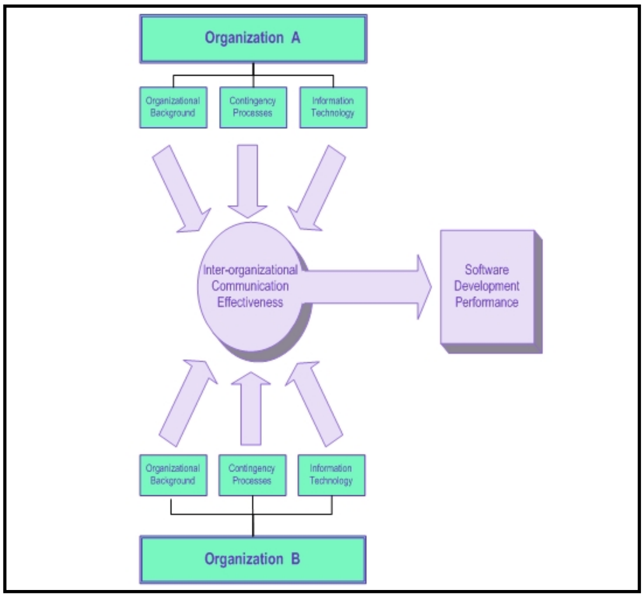

Figure 1 *: Conceptual Research Model*

Management contingency process/contingency policies cover sub-constructs which have been identified as influential factors for a success of collaborative ventures. We identify five sub-constructs for management contingency process: (1) Management of organizational behavior, (2) Team communication, (3) Software development practices, (4) Organizational support for activities, and (5) User involvement. Detailed discussions of these constructs are omitted here due to space limitation; the interested readers can find them in Mojgan et al., (2009)

Also studied in the first phase was the contribution of information technology factors, which have been grouped into (1) Group processes support and (2) Group communication support. We believe that many of these factors may overlap and indeed that there may be interaction effects among them. However, for the purpose of this research, we are not considering the interaction effect. Figure 1 shows the overall research model of this study.

The objective of the second phase is to discern whether the level of communication effectiveness between two collaborating organizations (represented as IPC here) affects the quality of software development process as well as the quality of the software produced.

Based on the literature review, the following eight main hypotheses were developed and a comprehensive data collection instrument was developed and used for investigation.

3.1 Organizational Background and IPC

H1: The IPC of an inter-organizational software development team is higher when they each have a higher level of organizational collaborative background.

H1a: The IPC of an inter-organizational software development team is higher when they have longer experience intheir core competency, collaboration and software development.

3.2 Information Technology and IPC

H2: The IPC of an inter-organizational software development team is higher for the organization with higher levels of IT support for communication and information exchange.

3.3 Contingency Processes and IPC

H3: The IPC of inter-organizational software development team is higher when they have higher levels of supporting management contingency processes for communication and information exchange in place.

3.4 IPC and the Performance of Software Development

H4: The performance of an inter-organizational software development team with a high level of IPC is more successful that one with a lower level of IPC.

H4a: The software development processes performance of an inter-organizational software development team with a high level of IPC is more successful that one with a lower level of IPC.

H4b: The software quality performance of an inter-organizational software development team with a high level of IPC is more successful that one with a lower level of IPC.

H4c: The user evaluation of software quality performance of an inter-organizational software development team with a high level of IPC is more successful that one with a lower level of IPC.

The research used an elaborate survey instruments based on 175 questions. Data were collected from collaborative software development teams located mostly in Northeastern United States. We submitted surveys and interviewed over 200 individuals in various functional areas of over 50 software development companies. We targeted to survey 100 projects developed through inter-organizational collaboration; but the completed surveys were received from only 84 projects. Of the remaining 16, there were 8 no-answers, and the surveys from the other 8 projects were unusable due to unanswered questions and/or missing pages from the questionnaires. A good majority of the 84 projects were small with a development budget of under $150,000.

Each project is evaluated by a manager, one to three technical staff, and one or more users of the software system. Consequently, three questionnaire instruments, one for each type of stakeholder, were developed. To minimize the effect of self-evaluation bias we have designed identical management and technical questionnaires, in content and format, so that one is answered from a management point of view and the other from the development point of view. In the final analyses, the answers from a same project were averaged and used in aggregate.

The selection of statistical methods, and the extent of the data analysis, was determined largely based on the nature and structure of the data. To determine whether to apply parametric or non-parametric studies, the assumption of normality was tested, and the results clearly indicated that the data violates the assumption of normality. Thus, it was decided to focus on statistical methods that did not require the normal distribution of the data. The analyses performed to test the hypothesis and explore the data were: confirmatory factor analysis (CFA), normality test, reliability test, validity test, and confirmation of the model fit.

5. Results and Analyses

Two software tools were used to analyze this data – SPSS for basic descriptive statistics and PLS Graph III to conduct the analyses. PLS is an appropriate method for exploratory research like this study [21]. First, the validity of the constructs was tested, following common methods for testing reliability and discriminant validity. Also, internal consistency or construct reliability was measured using the Cronbach's Alpha coefficient and the Fornell and Larcker's rho coefficient [22]. Second, the research model and the main hypotheses presented above were tested. Third, the mediating effect of IPC was tested. Since IPC is a new construct proposed in this study, which is modeled as a mediator, it is imperative to check if IPC is really a mediator. Other validation tests were performed are can be seen discussed in details [23].

The overall relationships among grand constructs were examined, offering an overall overview of the testing of hypothesis. To identify the relative weight and importance of the sub-constructs pertaining to individual constructs, the relationships among all sub-construct of input variables (predictors) and intervening latent variables IPC were evaluated.

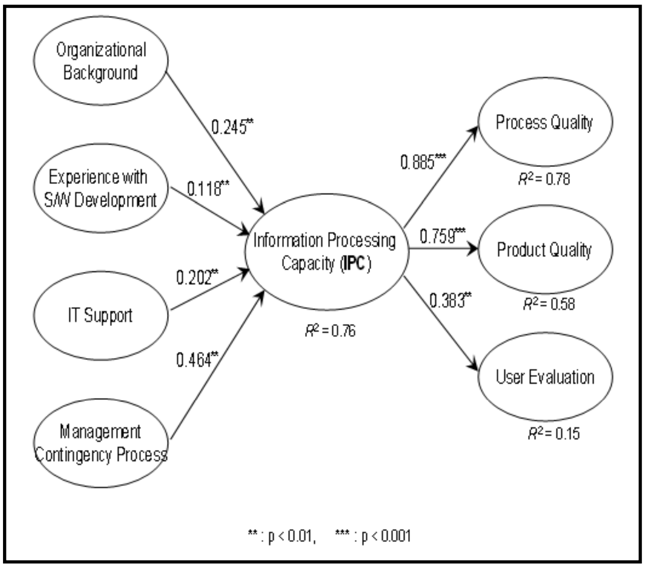

Figure 2 *: Results showing relationships in IPC*

Figure 2 displays the analysis results with grand constructs – organizational background, IT, management policies, IPC, and technical evaluation of software quality and process performance. The result shows that 76% of the variance in intermediate variable (IPC) is explained by independent constructs. Management contingency process has the greatest impact on IPC as it has the largest coefficient (0.464). Organizational background, IT support, and experience in software development follow. It also shows that 15% of variance in user evaluation, 78% of variance in software development process success and 58% of software quality success is explained by the intermediate variable (IPC). Table 1 depicts the path coefficients and their corresponding t-values. From Table 1 it is observed that all hypotheses are supported.

It is clear from these results that a good model fit is established with significant path coefficients with acceptably high R2 and internal consistency (i.e., construct reliability). We can also observe that hypothesis H1, H1a, H2, H3, H4a, H4b and H4c are all supported. Management explains 46%, while Organization background explains 25%, and IT is related to about 20% of variation in IPC.

The model was also tested in the sub-construct levels (not reported here). The detailed results are not included in this paper because of the space limitation; they are available with the authors upon request.

6. Conclusions

This study makes several contributions to the theoretical and practical aspects of inter-organizational collaborating in general and ICSD in particular. First, a new construct in collaboration, IPC, was defined and guidelines for its operations developed. Factors with significant influence on IPC and the success of inter-organizational software development were identified, assessed, and prioritized. Then, a model for the impact of IPC on the success of collaborative software development was created and tested using data from companies engaged in software engineering projects. It was also was created and tested using data from companies engaged in software engineering projects.

The result of this work also indicates that cultural familiarity and trust remain among the top four most influential factors in the success of inter- organizational collaboration, and that the standard software development practices may negatively influence some of the constructs under study. Consequently, management of alliances should deploy collaborative-aware practices. Finally, this study, which comprehensively examines inter-organizational collaboration, provides a baseline on which further investigation of inter-organizational collaboration can take place.

Table 1*: Path Analysis Results*

Hypothesis	Path Coefficient	T-statistics
H1: Organizational background → IPC	0.245	3.347**
H1a: Experience → IPC	0.118	3.307**
H2: IT support → IPC	0.202	2.745**
H3: Management contingency process → IPC	0.464	4.860**
H4a: IPC → Process quality	0.885	54.818***
H4b: IPC → Product (S/W) quality	0.759	22.228***
H4c: IPC → User evaluation	0.383	5.701**
* : p < 0.05, ** : p < 0.01, *** : p < 0.001 (one tailed)		

7. References

[1] Powell, A., Piccoli, G., and Ives, B. “Virtual teams: A review of current literature and directions for future research”, *The DATABASE for Advances in Information Systems* (35:1), pp 6-35, 2004.

[2] Kayworth, T.R., and Leidner, D.E. “Leadership effectiveness in global virtual teams”, *Journal of Management Information Systems* (18:3) pp 7-41, 2001-2002.

[3] Kraut, R.E., and Streeter, L. “Coordination in software development”, *Communications of ACM (38:3),* March. 1995.

[4] Boehm, B.W. “Software risk management: Priciples and practices”, *IEEE Software,* pp 32-41, 1991.

[5] George, J.A. “Virtual Best Practice”, in: *Teams Magazine,* pp. 38-45, 1996.

[6] Gorton, I., and Motwani, S. “Issues in cooperative software engineering using globally distributed teams,” *Information and Software Technology* (38:10), pp 647-656, 1996.

[7] Hoffmann, W.H., and Schlosser, R. “Success factors of strategic alliance in small and medium-sized

enterprises - An empricial survey", *Long Range Planning* (34), pp 357-381, 2001.

[8] Toffolon, C., and Dakhili, S. "A framework for studying coordination process in software engineering", *2000 ACM Symposium on Applied Computing,* Como, Italy, pp. 851-857, 2000.

[9] Rada, R., and Craparo, J. "Sharing standards: Standardizing software projects", *Communication of the ACM* (43:12), pp 21-25, December. 2000.

[10] Herbsleb, A.M. "Shared mental models, familiarity and coordination: A multi-method study of distributed software teams", *Proceedings of the 23rd International Conference on Information Systems (ICIS),* pp. 425-433, 2002.

[11] Espinosa, J.A., Kraut, R.E., Slaughter, S.A., Lerch, J.F., Herbsleb, J.D., and Mockus, A. *Shared mental Models, familiarity and coordination: A multi-method study of distributed software teams,* Proceedings of the 23rd International Conference of Information Systems (ICIS), pp. 425-433, 2002.

[12] Levina, N. "Collaborative practices in information systems development: A collective reflection-in-acion framework", *Proceedings of the 23rd International Conference on Information Systems (ICIS)*, pp. 267-277, 2002.

[13] Maybury, M. "Collaborative virtual environments for analysis and decision support", *Communication of the ACM* (44:12), pp 51-54, December. 2001.

[14] Leidner, D.E., and Kayworth, T. "A review of culture in information systems research: Toward a theory of information technology culture conflict", *MIS Quarterly* (30:2), pp 357-399, June. 2006.

[15] Handfield, R.B., and E. L. NIchols, J. *Introduction to Supply Chain Management,* 1999.

[16] Cramton, C.D. "Finding common ground in dispersed collaboration", *Organizatonal Dynamics* (39:4), pp 356-367, 2002.

[17] Massey, A.P., Hung, Y.-T.C., Montoya-Weiss, M., and Ramesh, V. "When culture and style aren't about clothes: Perceptions of task-technology 'fit' in global virtual teams", *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work,* pp. 207-213, 2001.

[18] Zeichick, A. "Easy Real Time Collaboration", in: *Information Week,* pp. 84-86, 1999.

[19] Marca, D.A., and Perdue, B.A. "A software engineering approach and tool set for developing Internet application", *Proceedings of the 22nd International Conference on Software Engineering,* ACM Press, Limerick Irelend, pp. 738-741, 2000.

[20] Qureshi, S., and Zigurs, I. "Global applications of collaborative technology, Paradoxes and prerogatives in global virtual collaboration", *Communications of the ACM* (44:12), 2001.

[21] Gefen, D.S., Detmar; Boudreau, Marie-Claude "Structural Equation Modeling and Regression: Guidelines for Research practices" *Communication of the AIS* (4:7), August. 2000.

[22] Fornell, C., and Larcker, D. "Structural equation models with unobservable variables and measurement error", *Journal of Marketing Research* (18:1) , pp 39-50, 1981.

[23] Mohtashami, M., Sylla, C., Il IM, and Deek, F.P. "A study of inter-organizational collaboration in software development", *International Journal of Information and Technology Management* (in press), 2009.